ORIGINAL RESEARCH

Cybercrime detection techniques based on support vector machines

Hanif Mohaddes Deylami, Prof. Yashwant Prasad Singh

Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

Correspondence: Hanif Mohaddes Deylami. Address: Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia. Telephone: 06-017-871-7244. Email: hmdeilami@gmail.com.

Received: February 23, 2012 **Accepted:** June 25, 2012 **Online Published:** November 19, 2012
DOI: 10.5430/air.v2n1p1 **URL:** http://dx.doi.org/10.5430/air.v2n1p1

Abstract
This paper presents the cybercrime detection model by using support vector machines (SVMs) to classify social network (Facebook) dataset. We try to compare between three kinds of classification algorithms such as: SVMs, AdaBoostM1, and NaiveBayes in order to find a high percentage of classification accuracy. Finally, we conclude SVMs as the best classification algorithm, which uses different breeds of kernel functions in order to improve the classification accuracy on Facebook dataset. Besides, we are using the Weka 3.7.4 software to evaluate classifiers on Facebook dataset.

Key words
Artificial Intelligence, Cybercrime, Support Vector Machine, Kernel Method, Facebook

1 Introduction
Nowadays, the users of Internet services are unthinkably increasing in the entire world. Internet services are the largest and richest source of information in the form of social networks, online applications, online videos and images, which lay among billions of web pages and blogs. The Internet users become connected to the networks for finding information, data, and friends. This situation makes suitable opportunities to hijack information and data by hackers and crooks. Cybercrimes currently is one of the most progressive offenses as problems in cyberspace, which involves any criminal act/transaction with computers and networks, such as: Spam, Drug Trafficking, Sales and Investment Fraud, Hacking, Cyber Terrorism, and Phishing [1, 2]. It is defined for the behaviours of an executable code by observing its usage dynamically. This paper presents Support Vector Machine (SVM) techniques in order to detect cybercrime in social network (Facebook) dataset.

SVM algorithm was first introduced in 1992, by Boser, Guyon, and Vapnik [3]. Support vector machines (SVMs) are a set of related supervised learning methods [4] deployed for regression and classification [5, 6]. Many software and application such as: Bioinformatics, Text Categorization, Ranking (e.g. Google search engine), Machine Vision, and Handwritten Character Recognition [7, 8], are designed using SVM algorithms in order to improve the percentage of classification accuracy. The fundamentals of SVMs have been enhanced by Vapnik [9] and became popular due to many promising features such as: better empirical performance. This method based on statistical learning theory, which is used to solve classification and regression problems [10]. The SVMs originally developed for binary classification problems, which uses the hyperplane to define separating decision boundaries among data points of different classes. "SVMs are learning algorithms that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory" [11]. SVMs are classifying linearly/nonlinearly input space by constructing hyperplane in feature space, which separates data optimally into two categories with different classes (see Figure 1) [10, 11].

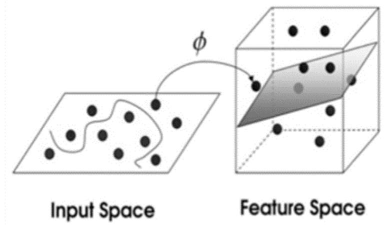

Figure 1. Separating Hyperplane in Feature Space SVMs [11].

The rest of this paper is organized as follows. Section 2 elaborate technical details about Support Vector Machines (SVMs) with mathematical backgrounds. Section 3 represents Cybercrime (Spam) Detection Model (CDM) and analysis the results of SVM, AdaBoostM1, and NaiveBayes on Facebook dataset. In addition, this section elaborates further experimental evolution is conducted with different kernel functions such as: Polynomial, RBF and Puk kernel functions to classify the dataset in order to improve the percentage of accuracy. Finally, in section 4 we conclude our research by comparing result of three kinds of classifiers.

2 Technical details about SVMs

Support Vector Machine (SVM) methods was developed to construct separating hyperplane for classification problems. SVM constructs functions (hyperplanes either in input space or in feature space) from a set of labelled training data. This hyperplane will try to split the positive samples from the negative samples. The split will be selected to have the largest distance from the hyperplace/hypersurface to the nearest of the positive and negative samples. Intuitively, this causes correct classification for testing data, which is near, but not equal to the training data. Throughout training phase SVM takes a data matrix as input data and labels each one of samples as either belonging to a given class (positive) or not (negative). SVM treats each sample in the matrix as a vector in a high dimensional feature space, where the number of attributes identifies the dimensionality of the space. Afterward, SVM learning algorithm determines the best hyperplane, which separates positive and negative training samples. Then, the trained SVM can be used to make predictions about a test sample's membership in the class. Briefly, nonlinear SVM requires mapping of the N dimensional input space into a high dimensional feature space. Finally, the linear classifier constructed in this high dimensional feature space [11, 12].

2.1 Maximum margin hyperplanes

The linear separable class includes many kinds of boundaries with different margins. The best margin (ρ) can be established by putting parallel decision boundaries with the closest distance from positive and negative samples, and then resides the hyperplane away from these positive and negative samples. As it is shown in Figure 1, there are many such hyperplanes. We assume the Facebook dataset is non-linearly separable and so there is no guarantee that the hyperplanes will act equally well on unseen samples (generalization). The classifier should choose one of these hyperplanes as its decision boundary, based on how well they are expected to perform on test samples. To elaborate how the different choices of the hyperplanes affect the generalization errors, consider the two decision boundaries B_1 and B_2 as it is shown in Figure 2. This Figure represents two decision boundaries B_1 and B_2, which both of these decision boundaries can separate the training samples into their respective classes without any misclassification errors. Each decision boundary B_i is related to a pair of hyperplanes, denoted as b_{i1} and b_{i2}, respectively. The b_{i1} is established by moving a parallel hyperplane away from the decision boundary until it obtains the closest negative samples (-), while b_{i2} is established by moving a hyperplane until it obtains the closest positive samples (+). The distance between these two hyperplanes in Figure 2 is named as the margin of the classifier. As it appears in Figure 2 the margin for B_1 is larger than for B_2, so B_1 considered being the maximum margin hyperplane of the training samples. This is the simplest kind of SVM (Called a Linear SVM).

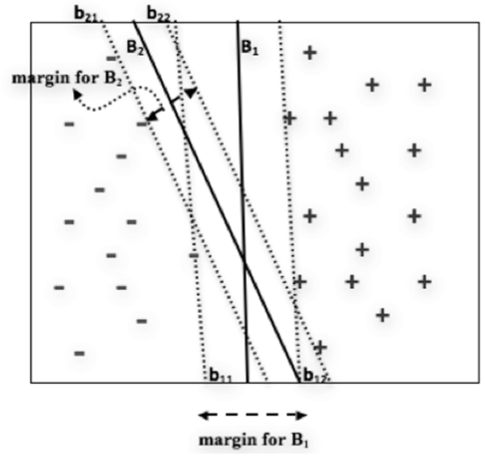

Figure 2. Margin of a Decision Boundary

2.2 Linear separable SVM classifier

A linear SVM classifier is a hyperplane with largest margin, which maximizes the geometric margin of training dataset. The linear SVM is one of the best solutions to multiclass classification problems. Figure 3 represent the mathematical background of linear SVM.

The dataset includes N training samples. Each sample is indicated by (x_i, y_i) and i=1, 2, …, N, where $x_i=(x_{i1}, x_{i2}, \cdots, x_{in})^T$ corresponds to the attribute set for the ith samples. Conventionally, let $y_i \in \{-1, 1\}$ be its class label. The decision boundary of a linear classifier can be written as follows:

$$W. x + w_0 = 0 \tag{1}$$

$W \in R^n$ and w_0 are the parameters of the pattern. Here "." is dot product. Figure 3 shows a two-dimensional training set, which a line applied as a decision boundary for separating the training samples into their respective classes. The training set $\{(x_i, y_i)\}$ i= 1, ..., N, $x_i \in R^n$, $y_i \in \{-1, 1\}$ be separated by a hyperplane with margin ρ. Then, for each training example (x_i, y_i):

$$\begin{cases} w^T x_i + b \leqslant -\rho/2 & \text{if } y_i = -1 \\ w^T x_i + b \geqslant \rho/2 & \text{if } y_i = 1 \end{cases} \Leftrightarrow y_i\,(w^T x_i + b) \geq \rho/2 \tag{2}$$

The inequality (2) is appropriate for any support vector x_s. Thus, we rescale w and b by ρ/2, and obtain that distance between each x_s and hyperplane as follows:

$$r= y_s\,(w^T x_s+b)/\|w\|= 1/\|w\| \tag{3}$$

Then, the margin can be expressed through (Rescaled) w and b as:

$$\rho = 2\,r = 2/\|w\| \text{ and } \|w\| := \sqrt{(x_1^2+\ldots+x_n^2)} \quad (4)$$

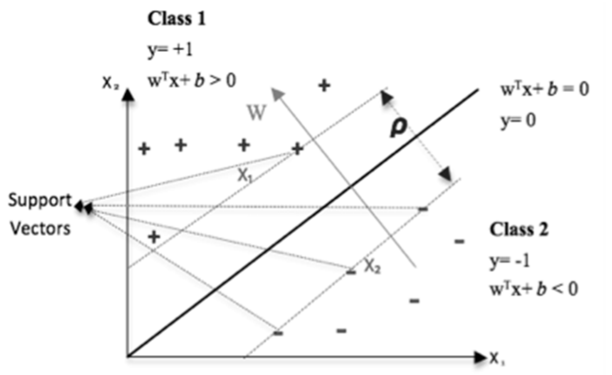

Figure 3. Linear Separating Hyperplanes

2.3 Non-linear SVM classifier

In nonlinear classifier design, the SVMs algorithm maps input data into a high dimensional feature space, and then construct a hypersurface in input space. The mathematical background is:

$$x \rightarrow \Phi(x) \quad (5)$$

and then, learn the map from Φ(x) to y is:

$$F(x) = w.\Phi(x) + b \quad (6)$$

Then, the non-linear SVMs try to classify the dataset by using kernel functions for mapping data into a higher dimensional space (see Figure 4).

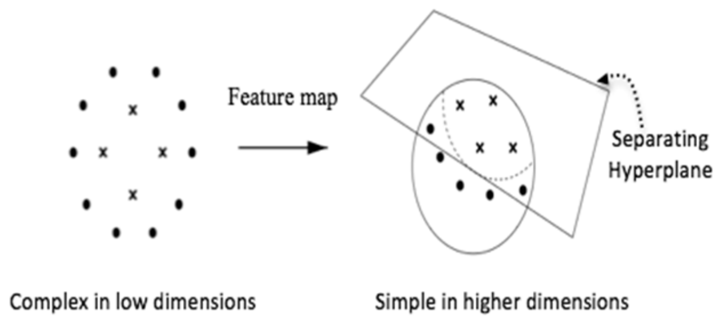

Figure 4. Non-Linear SVM Feature Map [6]

2.4 Mapping to a High-Dimensional Space

The SVMs for handling classification problems try to use the n-dimensional data point's x_i (i= 1, 2, …, N) in order to change non-linear function in complex low dimension to linear function in simple high dimension in feature space [11, 13].

2.5 Kernel Methods

Kernel Methods (KMs), by using kernel function try to find a relation between data in a dataset. The KMs are mapping data into a higher dimensional space by choosing the appropriate kernel function in order to improve classification accuracy [14].

2.5.1. Kernel trick

In SVM the optimal hyperplane is defined to maximize the generalization ability. But, if the training data are not linearly separable, the achieved classifier may not have high generalization ability. Thus, to enhance linear separability, the original input space is mapped into a high-dimensional dot-product space called the feature space. Kernel trick is assisting SVM algorithms to maximize the generalization ability. According to the nonlinear separable space, a mapping function from n-dimensional input space to l-dimensional feature space is defined as [15]:

$$g(x) = (g_1(x), g_2(x), \ldots, g_i(x))^T \quad \text{For} \quad i=1, 2, \ldots, N \tag{7}$$

Besides, the linear function as decision boundary in the feature space is defined as:

$$D(x)= w^T g(x) + w_0 \tag{8}$$

In equation (8), w is an l-dimensional vector and w_0 is a bias term. Then, we use kernel techniques for dealing with inner product in the high dimensional feature space. These techniques are called “Kernel Trick”. The kernel function appears as dot product (i.e.: $x_i.x_j$ or $y_i.y_j$) in some corresponding feature space and then maximize margin subject to [15]:

$$\begin{cases} \sum^{N} \alpha_i - \frac{1}{2} \sum^{N} \alpha_i \alpha_j y_i y_j x_i.x_j \\ C \geq \alpha_i \geq 0, \sum \alpha_i y_i = 0, \qquad \text{for } i=1, 2, \ldots, N \end{cases} \tag{9}$$

and then, the kernel function can be introduced as an equation (10):

$$K(x_i , x_j)= \phi (x_i). \phi (x_j) \tag{10}$$

2.5.2. Kernel functions

The main idea of SVMs is to map the original data points from the input space to a high dimensional feature space, or even infinite dimensional, since the classification problem becomes simpler in the feature space. This mapping will be done by an appropriate selection of kernel functions, which means, the key features of SVMs are use kernel functions such as: Linear Kernel, Polynomial Kernel, Radial Basis Function Kernel (RBF), and Puk kernel [13, 16, 17].

- Linear Kernel

The linear kernel function is the simplest kind of kernel functions. If a classification problem is linearly separable in the input space, we don’t require mapping the input space into a high-dimensional space. The linear function with inner products <x, z> and optional constant “c” is specified as follows:

$$K(x, z)= x^T z + c \tag{11}$$

- Polynomial Kernel

The polynomial Kernel uses to normalize training data. The polynomial kernel with degree d, where d is a natural number, is specified as follows:

$$K(x, z) = (x^T z + c)^d \tag{12}$$

- Radial Basis Function Kernel

Radial Basis Function (RBF) kernel is similar with RBF in neural networks. The RBF kernels hold the Euclidean distance, which are not robust to outliers. To achieve this, Chen [18] proposed the M-estimator-based robust kernels, which are more robust versions of RBF kernels, presenting the idea of robust statistics. The radial basis function (RBF) kernel is specified by:

$$K(x, z) = \exp(- \| x - z \|^2 / (2\sigma^2)) \tag{13}$$

- Puk Kernel

The Puk Kernel function proposed by B. Ustun et al. [19] to solve the SVMs regression problems. In Puk kernel function all distances between input samples and map weights are in the range of [0, 1] and try to normalize the distance by changing the value of w and σ between two vectors. The parameter w controls the actual shape, which means the tail of the function. The parameter σ determines the width of the Pearson VII function. It is specified as follows:

$$K(x_i, x_j) = 1 / [1 + ((2 \sqrt{} \|x_i - x_j\|^2 \sqrt{} 2^{1/w} - 1) / \sigma)^2]^w \tag{14}$$

2.6 Binary classification tree support vector machine

Binary classification tree in SVM (BCT-SVM) is one of the significant classification techniques in machine learning. Fundamentally, SVMs extend for solving binary classification problem by using hyperplane with the maximum margin. Binary classification tree support vector machine is the task that consists of two groups of samples. It means, a dataset construct by a group of positive/cybercrime samples and another group includes set of negative/non-cybercrime samples (e.g. Medical testing: the instance is disease or no-disease). BCT-SVM is consisting K-1 internal nodes and k terminal nodes. For building classification tree, each internal node is putting in the right child or left child node. Then, each internal node is used to find the optimal construction of the classifier.

2.7 Sequential minimal optimization algorithm

Sequential Minimal Optimization (SMO) algorithm is used for solving large quadratic programming (QP) optimization problems widely deployed for the training of SVMs [20]. SMO breaks these large QP problems into a series of smallest possible QP problems, and then these small QP problems are solved analytically, which prevents using a time-consuming numerical QP optimization as an inner loop. It includes two parts: training and mapping such that the training builds the map from a high dimensional input space to a lower dimensional space with input samples. SMO is fastest for linear SVMs and sparse data sets; due to SMO's computation time is dominated by SVM evaluation.

3 Experimental and comparison with other classifiers

This section presents the experimentation on Facebook dataset and also comparison with other classifier algorithms such as: AdaBoostM1, and NaiveBayes. After that, we try to optimize the percentage of classification accuracy by using different kinds of kernel functions such as: Polynomial kernel, Radial Basis Function (RBF) kernel, and Puk kernel.

3.1 Dataset and system requirements

This paper considers Facebook-Dataset, which is taken from the Max Plank institute for software systems (http://socialnetworks.mpi-sws.org/data-wosn2009.html). The size of the dataset is 57kB. Additionally, the dataset includes two classes with six attributes are listed in Table 1. Moreover, the number of samples is 2700, which divided into 2430 instances (Training data= (2700 ⁄ 10) ×9)) as a training data and 270 instances (Test data= 2700 - 2430) as a testing data (see Table 2). The system requirements are Weka 3.7.4 software on Mac OS X version 10.6.8 with processor 2.4GHz Intel Core 2 Duo, memory 4GB, and 1067 MHz DDR3.

Table 1. Facebook Dataset Attributes

#	Attribute	Nominal	Numerical
x1	FirstUser	-	✓
x2	SecondUser	-	✓
x3	Freq_Message	-	✓
x4	Freq_Post	-	✓
x5	Freq_Business	-	✓
x6	Freq_Application	-	✓

Table 2. Information of Facebook Dataset

Dataset	#Sample	#Class	#Attribute
Training Samples	2430	2	6
Testing Samples	270	2	6

3.2 Classifiers and confusion matrix

Each dataset with N classes (N⩾0) can be creating the N×N confusion matrix. The confusion matrix uses to measure classifier performance. Each cell in confusion matrix has a specific meaning, which refers to the feature of dataset. The True Positive (TP) rate is the proportion of samples which were classified as class a, among all samples which truly have class a, i.e. How much part of the class was captured. It is equivalent to Recall. In the confusion matrix, this is the diagonal element divided by the sum over the relevant row. The False Positive (FP) rate is the proportion of samples, which were classified as class a, but belong to a different class, among all samples, which are not of class a. In the matrix, this is the column sum of class a minus the diagonal element, divided by the row sums of the class b. The Precision is the proportion of the samples, which truly have class x among all those, which were classified as class x. In the matrix, this is the diagonal element divided by the sum over the relevant column. The F-Measure is simply equal to 2×Precision×Recall/ (Precision+Recall), a combined measure for Precision and Recall. These measures are applicable for comparing classifiers.

3.3 Cybercrime detection model (CDM)

SVM by using kernel functions try to improve the computational of complexity, and over-fitting problems in order to classify dataset with the maximum margin hyperplane. In this section, we elaborate the proposed SVM model for spam detection on Facebook dataset. To distinguish spam nodes from normal users in Facebook, four features of each user are extracted in this paper. Extracted features are listed as follows: (1) Freq_Message, (2) Freq_Post, (3) Freq_Business, and (4) Freq_Application. Furthermore, we elaborate three states between two users can be happened are listed as follows: (5) communication reciprocity (CR), (6) communication interactive average (CIA), and (7) clustering coefficient (CC). Instead, we utilize features proposed by previous works and focus on devising an effective model that is more robust in social networks (Facebook). The main idea of extracting features from the Facebook dataset is that based on domain knowledge, the behaviour of spammers is quite different from that of normal users. Thus, many researchers attempt to model the unusual behaviour as features to block spams. Two classes, with three relevant states, and totally four features are included in our model. The details are depicted as follows:

- Freq_Message, Freq_Post, Freq_Business, and Freq_Application. For each node in a network, the features Freq_Message, Freq_Post, Freq_Business, and Freq_Application are the number of messages, posts, businesses, and applications data, which received (sent) by this node, respectively. Regarding to the Facebook dataset analysis, since only a very small number of users will reply spams, the nodes of spammers usually have near-zero values. On the other hand, spams nodes will have much higher sent values than normal users.

- Communication Reciprocity (CR) and Communication Interactive Average (CIA). It is observed that data interaction between normal users is frequent, meaning that when a user receives a data, he/she will usually reply this data to the sender. On the other hand, only a few users will reply data's to spammers. Based on this observation, the both of the communication reciprocity (CR, first proposed in [21]) and communication interactive average (CIA, first proposed in [22]) are presented to capture this difference. CR is defined as follows:

  $$CR\ (x) = (\ |OS(x) \cap IS(x)|\ ) \ /\ |\ OS(x)\ | \tag{15}$$

  Where OS(x) is the set of nodes that have ever received a data from node x, and IS(x) is the set of nodes that have ever sent a data to node x. The feature CR measures the ratio of nodes that have data interactions with node x. On the other hand, CIA is defined as follows:

  $$CIA\ (x_i) = (\sum w(e_{ji})/w(e_{ij}))\ /\ Out\_data\ of\ x_i \tag{16}$$

  CIA differs from CR in the numerator of the equation. CIA further considers whether each data has the feedback. The $w(e_{ij})$ denotes the number of data sent from node x_i to x_j. The $w(e_{ji})$ denotes the number of feedbacks from node x_j to x_i. It is apparent that the larger the CR and CIA values are, the less suspicious for a node to be a spammer.

- Clustering Coefficient (CC). Clustering coefficient (CC) is first introduced [23]. In this paper, we consider the clustering coefficient for undirected graphs. CC is defined as follows.

  $$CC\ (x_i) = n_{xi}\ /\ ([Z_{xi}\ (Z_{xi} - 1)]\ /\ 2) \tag{17}$$

  Where Z_{xi} is the number of neighbours of x_i, and n_{xi} is the number of existing links between the neighbours of x_i. It can be imagined that the data of Facebook formed by a group of normal users is expected to possess more links. The feature CC is designed to catch this phenomenon and these links can be viewed as the Friends-Of-Friends (FOF) relationship. Since spammers usually spread a great quantity of spams randomly, the CC values of normal users are generally much larger than those of spammers.

The following procedure shows the Cybercrime Detection Model (CDM) by using SVM in seven steps:

```
Procedure CDM
  Input:
    Facebook Dataset;
    A threshold max_inst;
Preprocessing:
  An SVM model:
      1. Choose the pure nodes that have sent only harms as training instances;
      2. If (the number of pure nodes > max_inst)
      3. Do the suitable classification sampling to make the number of training instances equal to max_inst;
      4. Scale each feature value to [0, 1];
SVM Model:
      5. Perform the SVM training, testing, and SMO;
    6. Return to SVM model;
  Output:
    7. Cybercrime/No Cybercrime;
  End.
```

Initial SVM training

After the data Facebook is constructed with the training set and the feature values of each user are extracted, the proposed procedure CDM (Standing for Initial SVM Training) is performed to train the initial SVM model. The algorithmic form of procedure CDM is shown in section 3.3. Initially, the pure nodes that have sent only spams are chosen as training instances. This step ensures that the labels of choosing training instances are assured. Moreover, if the number of pure nodes is still larger than a predefined threshold max_inst, we perform the proportionate stratified sampling to limit the maximum number of instances to max_inst. The proportionate stratified sampling retains the proportion of class labels. Afterward, each feature is scaled to the range of 0 to 1.

At this stage, the preparation processes before the SVM training are finished. The CDM procedure employs the well-known LIBSVM library [24] to train the SVM model. Moreover, for the application of Facebook spam detection, the false positive error is much more severe than the false negative error. Thus, we set that the penalty of false positive error is 10 times than the penalty of false negative error. Other parameters are set as default values of LIBSVM.

3.4 Experimental evaluation on facebook dataset

To find the high percentage of classification accuracy, we conduct several experiments to explore the performance of classification and detection results. Therefore, we try to classify the Facebook dataset with different algorithms such as: SVM, AdaBoostM1, and NaiveBayes by using different kinds of kernel functions. We use the Weka 3.7.4 software for experimenting on Facebook dataset. Furthermore, we are changed the Weka software filter samples into unsupervised learning in order to let to an algorithm for deciding to which group of data are useful and then, changed the attribute of filters to discretize to change the range of numeric attributes into nominal attributes. The test option is cross-validation with ten folds. Then, the percentage of correctly classified samples represents, which algorithm appropriate for our study.

Firstly, we deployed the SMO (Sequential Minimal Optimization) as a kind of SVMs algorithm with non-linear dataset and obtained associated results as shown in Table 3. Therefore, the correctly classify is 98.5556 percent of all samples. That means, 2661 samples correctly classified and 39 samples incorrectly classified from all samples (2700) of the training set. The number of support vectors is 104, as well as time is taken for building model according to training data is 0.28 seconds. Here we have two classes, and therefore our confusion matrix is 2×2. The number of correctly classified samples is the sum of diagonals in the matrix; all others are incorrectly classified (Class “a” gets misclassified as “b” is zero instance and class “b” gets misclassified as “a” is 39 instances). The True Positive (TP) rate for class “a” is 2661/(2661+0)= 1 and for class “b” is 0/(0+39)= 0. The False Positive (FP) for class “a” is 39/(39+0)= 1 and for class “b” is 0/(0+2661)= 0. Moreover, the Precision of class “a” is 2661/(2661+39)= 0.98 and for class “b” is 0/(0+0)= 0.

Table 3. Comparison between Different Classifier Functions

Classifier	Accuracy (%)	Consuming Time (s)	#Correctly Classify Instance	#Incorrectly Classify Instance
SVM	**98.5556**	**0.28**	**2661**	**39**
AdaBoostM1	98.5185	0.21	2660	40
NaiveBayes	98.3333	0.04	2655	45

Secondly, we deployed the AdaBoostM1 algorithm, which represents the performance of the classifier, be close to SVM algorithm. The AdaBoostM1 algorithm uses different kinds of classifier such as: Decision Stamp, J48, Random Forest, and Random Tree with different number of performance iteration for evaluating classification accuracy of dataset. The percentage of correctly classify is 98.5185 of all samples (2700). It means, the number of correctly classified of the training set is 2660 samples and the number of incorrect classifications is 40 samples. Besides, the time is taken for building model is 0.21 seconds.

Thirdly, we deployed the NaiveBayes algorithm by using different kinds of kernel estimator to classify dataset. The classifier in NaiveBayes is not an updatable classifier and it uses a default precision of 0.1 for numeric attributes. Thus, the experimental evaluation on the test set by the NaiveBayes classifier represents the weak percentage of accuracy in comparison with SVMs method. Moreover, the number of correctly and incorrectly classification instances is 2655, and 45 samples, respectively. The correctly classify is 98.3333 percentage of all samples, as well as, the time is taken for building model is 0.04 seconds.

Finally, Table 3 illustrates the comparison between SVM classifier in contrast to the AdaBoostM1 and the NaiveBayes classifier. The obtained result shows that the SVM classification accuracy is better than these classifiers, but it needs more optimization of parameters to enhance the performance on unseen samples. However, the consuming time in AdaBoostM1 algorithm is better than the other algorithms, but the significant feature in this step is the percentage of accuracy.

To continue, we use different kinds of kernel functions such as: Polynomial kernel, RBF (Radial Basis Function) kernel, and the Puk kernel in order to improve the percentage of accuracy of SVM algorithms. At first, we deployed polynomial kernel with different value of C, which is shown in Table 4. It represents the accuracy of SVM by polynomial kernel (C=10) is 98.5556 percent. Moreover, the time taken for building model is 0.28 seconds and the number of support vectors is 104. Besides, in the second step we are using the RBF kernel function, which applied to our dataset (see Table 4). The accuracy of classifier with C=10 is 98.5556 percentage. Furthermore, the number of support vectors in this function is 80, and the time is taken for building model is 0.36 seconds. Thus, the percentage of accuracy is same as polynomial kernel function but it constructs a model slower than a polynomial kernel function. The third one is SVM with Puk kernel function and different value of C, which ability to behave as a generic kernel. The high accuracy with best computational time to build model is 0.47 seconds when the value of C is 10. Finally, Table 5 represents the best results of the SVM algorithm with different kinds of kernel functions and value of C. However, the experimenting results on Facebook dataset represents the polynomial kernel function with C equal ten gave the best value to improve accuracy and provide a possible way to handle over-fitting problem in SVMs.

Table 4. Comparison between Kernel Functions with Different Value of C

Classifier	Kernel	C	Accuracy (%)	# Support Vectors	# Kernel Evaluation	Computation Time (s)
SVM	Polynomial*	**10**	**98.5556**	**104**	**282866**	**0.28**
		30	98.5556	121	331679	0.49
		50	98.5556	143	393050	0.31
		70	98.5556	180	512539	0.63
		90	98.5556	213	620046	0.66
	RBF	**10**	**98.5556**	**80**	**215580**	**0.36**
		30	98.5556	79	215788	0.41
		50	98.5556	82	220835	0.46
		70	98.5556	81	220935	0.51
		90	98.5556	90	244846	0.41
	PUK	**10**	**98.5556**	**133**	**367731**	**0.47**
		30	98.5556	205	598117	0.58

50	98.5556	175	515114	0.55
70	98.5556	193	587252	0.58
90	98.5556	195	582052	0.58

*Degree of Polynomial (d) is 1.0.

Table 5. The Best Result in Each Kernel Function

Kernel	C	Accuracy (%)	# Support Vectors	# Kernel Evaluations	Time Consuming (s)
Polynomial	**10**	**98.5556**	**104**	**282866**	**0.28**
RBF	10	98.5556	80	215580	0.36
PUK	10	98.5556	133	367731	0.47

4 Conclusions

In this paper, we design a complete cybercrime (spam) detection model (CDM), which is used to improve the classification accuracy of Facebook dataset. Moreover, we deploy SVM, AdaBoostM1 and NaiveBayes classifier on the Facebook dataset, which represents the SVM is efficient and effective in compare of AdaBoostM1 and NaiveBayes. Besides, we try to improve the SVM accuracy by applying different kernels. Then, we are founding the polynomial kernel with optional constant equal ten (C=10) yielded the high percentage of classification accuracy than other kernel functions such as: Radial Basis Function (RBF) kernel and Puk kernel.

References

[1] Lane, Randal and Ashraf. “Hybrid Intelligent Systems for Network Security”. 2011.

[2] Taiwo Oladipupo Ayodele. “Type of Machine Learning Algorithm”. 2010.

[3] Vapnik, V., Boser, B. E., Guyon, I. A Training Algorithm for Optimal Margin Classifiers. Conference on Computational Learning Theory 15th (COLT). 1992; 144-152.

[4] V.N.Vapnik. “The Nature of Statistical Learning Theory”. 1995.

[5] Chan, A. K., XU, P. Support vector machine for multi-class signal classification with unbalanced samples. International Joint Conference on Neural Networks. 2003; 1116-1119.

[6] Xiaojin Zhu, Andrew B. Goldberg. “Introduction to Semi-Supervised Learning”. 2003.

[7] Nello Cristianini, John Shawe-Taylor. “An introduction to support vector machines: and other kernel-based learning”. 2000.

[8] Chen, Lin. “Working set selection using second order information for training SVM”. 2005

[9] Vapnik, V.N. The Nature of Statistical Learning Theory, Second Edition, New York: Springer. 2000.

[10] Tzu-Yen, Chin-Hsiung and Chu-Cheng. “Virus Prevention Model Based on Static Analysis and Data Mining Methods”. 2009.

[11] Cristianini, N., Shawe Taylor, J. “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods”. 2000.

[12] Christopher J.C. Burges. “A Tutorial on Support Vector Machines for Pattern Recognition”. 1998.

[13] Chi-Yao and Ming-Syan. “Incremental SVM Model for Spam Detection on Dynamic Email Social Networks“. 2009.

[14] Klaus-Robert Müller, and et al. “An Introduction to Kernel-Based Learning Algorithms”. 2001.

[15] Abe, S. Support Vector Machines for Pattern Classification. London: Springer-Verlag. 2010. http://dx.doi.org/10.1007/978-1-84996-098-4

[16] Genton, Marc G. “Classes of Kernels for Machine Learning: A Statistics Perspective”. 2001.

[17] César Souza. “Kernel Function for Machine Learning Application”. 2010.

[18] Chen, R. C., and et al. Detecting Credit Card Fraud by Using Questionnaire-Responded Transaction Model Based on Support Vector Machines. 2004; 800-806. Berlin: Springer-Verlag.

[19] Üstün, B., Melssen, and et al. Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel. Chemometrics and Intelligent Laboratory Systems. 2006; 81: 29-40. http://dx.doi.org/10.1016/j.chemolab.2005.09.003

[20] Platt, C. J. Fast training of support vector machines using sequential minimal optimization Source. In Advances in kernel method: support vector learning. 1999; 185-208.

[21] L. H. Gomes, R. B. Almeida, et al. Comparative graph theoretical characterization of networks of spam and legitimate email. Proc. of the 2nd Conference on Email and Anti-Spam (CEAS). 2005.

[22] H.-Y. Lam and D.-Y. Yeung. A learning approach to spam detection based on social networks. Proc. of the 4th Conference on Email and Anti-Spam (CEAS). 2007.

[23] D. J. Watts and S. Strogatz. Collective dynamics of “small-world” networks. Proc. of Nature. 1998.

[24] C.-C. Chang and C.-J. Lin. 2001. LIBSVM: a library for support vector machines. Available from: http://www.csie.ntu.edu.tw/~cjlin/libsvm.